Intermec®

VERDEX

Mobile Component

User's Guide

Intermec Technologies Corporation

Worldwide Headquarters
6001 36th Ave.W.
Everett, WA 98203
U.S.A.

www.intermec.com

Document Change Record

This page records changes to this document. The document was originally released as Revision 001.

Version Number	Date	Description of Change
002	5/2011	Revised to support software release 1.10. Added support for 70 Series computers. Revised content and descriptions of XML attributes and sample code.

Contents

Before You Begin

This section provides you with safety information, technical support information, and sources for additional product information.

Global Services and Support

Warranty Information

To understand the warranty for your Intermec product, visit the Intermec web site at **www.intermec.com** and click **Support** > **Returns and Repairs** > **Warranty**.

Disclaimer of warranties: The sample code included in this document is presented for reference only. The code does not necessarily represent complete, tested programs. The code is provided "as is with all faults." All warranties are expressly disclaimed, including the implied warranties of merchantability and fitness for a particular purpose.

Web Support

Visit the Intermec web site at **www.intermec.com** to download our current manuals (in PDF).

Visit the Intermec technical knowledge base (Knowledge Central) at **www.intermec.com** and click **Support** > **Knowledge Central** to review technical information or to request technical support for your Intermec product.

Telephone Support

In the U.S.A. and Canada, call **1-800-755-5505**.

Outside the U.S.A. and Canada, contact your local Intermec representative. To search for your local representative, from the Intermec web site, click **About Us** > **Contact Us**.

Service Location Support

For the most current listing of service locations, go to **www.intermec.com** and click **Support** >**Returns and Repairs** > **Repair Locations.**

For technical support in South Korea, use the after service locations listed below:

AWOO Systems
102-1304 SK Ventium
522 Dangjung-dong
Gunpo-si, Gyeonggi-do Korea, South 435-776
Contact: Mr. Sinbum Kang
Telephone: +82-31-436-1191
E-mail: **mjyun@awoo.co.kr**

IN Information System PTD LTD
6th Floor
Daegu Venture Center Bldg 95
Shinchun 3 Dong
Donggu, Daegu City, Korea
E-mail: **jmyou@idif.co.kr** or **korlim@gw.idif.co.kr**

Who Should Read This Manual

This User's Guide is for software developers who want to include VERDEX functionality in applications. This document provides you with information about the VERDEX Mobile Component, and how to install, configure, and operate the software.

Before you use the VERDEX Mobile Component, you should be familiar with your Intermec computer and with general concepts for application design.

Related Documents

The Intermec web site at **www.intermec.com** contains our documents (as .pdf files) that you can download for free.

To download documents

1. Go to **www.intermec.com** and click the **Products** tab.
2. Using the **Products** menu, navigate to your product page. For example, to find the CN50 computer product page, click **Computers** > **Handheld Computers** > **CN50**.
3. Click the **Manuals** tab.

For other products, click **Support** > **Manuals**. Use the **Product Category** field, the **Product Family** field, and the **Product** field to help you locate the documentation for your product.

About the VERDEX Mobile Component

The Intermec VERDEX Mobile Component application runs on selected Intermec mobile computers and provides Optical Character Recognition (OCR) on images taken of machine printed postal address labels. Address information in the images is compared with a database to verify the addresses and provide corrections when needed.

You can print a PDF417 shipping label to an Intermec PB22, PB32, or PB50 mobile printer (Intermec Fingerprint only) connected through Bluetooth.

This User's Guide explains how to install and use the Mobile Component, and how to call the application from another application so it can be integrated into your workflow.

Supported Computers

- 70 Series Mobile Computer with EA30 imager
- CN3 Mobile Computer with EA11 imager
- CN50 Mobile Computer with EA11 or EA21 imager

What's New

Version 1.10 of the Mobile Component includes support for the 70 Series computers with an EA30 imager.

Downloading and Installing the VERDEX Mobile Component

1. Go to **www.intermec.com** and choose **Support** > **Downloads**.
2. In the Product Category list, choose **Software & Tools**.
3. In the Product Family list, choose **Solution Components**.
4. In the Product list, choose **VERDEX** and then click **Submit**.
5. In the Applications list, click **VERDEX Solution Component** and follow the prompts to download and run the setup file. You need to log in to download files.
6. After the download and installation is complete, browse to C:\Program Files\Intermec\VERDEX\Runtime\WCE600\WM6\Armv4i and locate the VERDEX.cab file.

7 Copy VERDEX.cab to the Cabfiles folder on the mobile computer.

8 Warm boot the mobile computer. The VERDEX installation begins and takes a few minutes to complete. You do not need to warm boot the computer again after the installation is complete.

   For more information about warm booting the mobile computer, see the computer user's manual.

About Licensing

To use the full VERDEX functionality, you need to purchase a run-time license from Intermec. One license is required for each computer using VERDEX. A valid run-time license unlocks postal address verification, as well as access to features and databases that may be available only with that license.

For development and testing, VERDEX ships with demonstration databases you can use without the run-time license.

Although databases you provide do not require a license for access, some databases require a separate license.

To order a license or for more information, contact your Intermec sales representative.

About the Sample Addresses

If you are running the Mobile Component in demo mode, print the sample addresses to use when verifying address data. When you install VERDEX to the default location, the sample address files are located in the C:\Program Files\Intermec\VERDEX\Documents directory:

- DemoAddresses_DEU.pdf (use with the sample Germany database)
- DemoAddresses_GBR.pdf (use with the sample United Kingdom database)
- DemoAddresses_USA.pdf (use with the sample United States database)

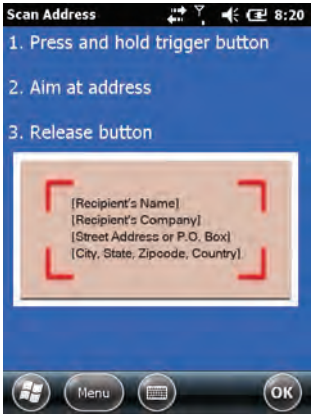
By default, the Mobile Component uses the sample United States database.

Reading and Verifying Addresses

To use the Mobile Component to read and verify addresses in demo mode, you need to print at least one of the sample address files. For this procedure, use the DemoAddresses_USA.pdf file. For more information, see the previous section, **"About the Sample Addresses" on page 10.**

To read and verify addresses

1. Tap **Start** > **Programs** > **VERDEX Demo**. After the application initializes, the Scan Address screen appears.



2. Press and hold the **Scan** button. The aiming beam or aiming line turns on and stays on.

3 Center the aiming beam around one of the sample addresses. For the 70 Series, position the aiming line just below the bottom line of the address.

Reading an Address: *This example shows a CN50 computer.*

Note: In general, for best results, the larger the address font size, the further away from the label you should hold the computer.

4 Release the **Scan** button. The imager reads the address, and VERDEX compares the address to the database.

When VERDEX is done processing the image, the VERDEX Results screen appears. For descriptions of possible VERDEX Results, see the next section, **"About VERDEX Results" on page 13.**

5 (Optional) Print a PDF417 address label. For more information, see **"Printing Address Labels" on page 18.**

6 Repeat Steps 2 through 4 to scan more addresses.

7 Tap **OK** to close the application.

About VERDEX Results

After VERDEX processes the image, the VERDEX Results screen appears with one of three possible messages.

VERDEX Results: Match Found

If the address is verified and matched exactly, the VERDEX Results screen appears with a "Match found" message:

To learn about printing an address label for the matched address, see **"Printing Address Labels" on page 18.**

VERDEX Results: Matches Found

If at least one possible match is found, this screen appears with a "Matches found" or "Partial Match" message:

In demo mode, if more than one match was found, tap ◀ and ▶ to view the possible matches, and tap **Select** to choose the currently displayed address.

To learn about printing an address label for the matched address, see **"Printing Address Labels" on page 18.**

VERDEX Results: No Match Found

If the computer could not read the address, or was unable to verify it against the database, this screen appears with a "No match found" message. A "?" character appears in the address in each position that VERDEX was unable to process.

You can also try entering the address manually as described in **"Manually Entering an Address" on page 16.**

Viewing the Scanned Image

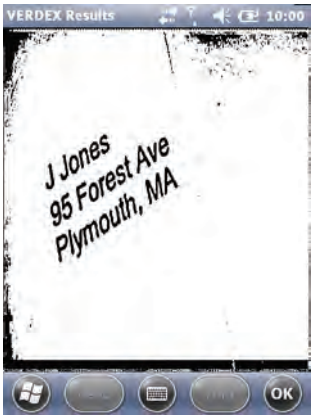
If VERDEX is unable to process an image, you can view the scanned image to get a better idea of how to improve the image quality, such as by framing the address differently or by holding the imager at a better angle.

To view the scanned image

1 Perform Steps 2 through 4 of **"Reading and Verifying Addresses" on page 11.**

2 Tap the thumbnail image in the upper right corner of the VERDEX Results screen. The scanned image appears.

Thumbnail in VERDEX Results Screen: *Tap the thumbnail to see the full scanned image.*



Scanned Image

3 Tap the image to return to the VERDEX Results screen.

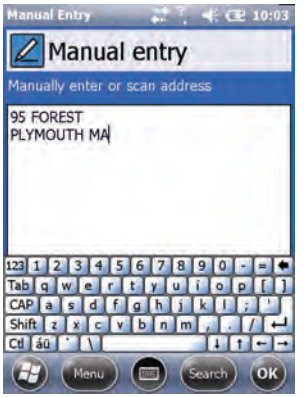
Manually Entering an Address

If the computer cannot read the address label, you can manually enter the address to verify it.

1 Tap **Menu** > **Manual Entry**.

2 Enter the address with the computer keypad, or tap  to use the onscreen keyboard.

Note: For best results, enter characters for country-specific postal codes using the standard for that country. For example, postal codes for the United Kingdom use uppercase letters.



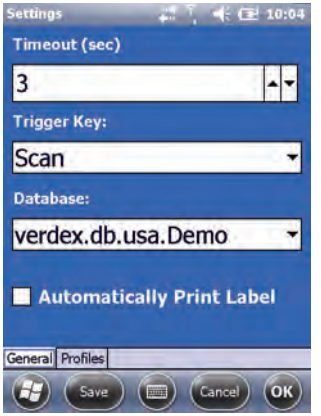
3 Tap **Search** to verify the address. The VERDEX Results screen appears.

4 Continue with Step 4 of **"Reading and Verifying Addresses" on page 11.**

Note: From the Manual Entry screen, you can also try scanning the address as described in Steps 2 and 3 of **"Reading and Verifying Addresses."**

Configuring Settings

1 Tap **Menu** > **Settings**. The Settings screen appears with the General tab selected.



2 Change settings as follows:

- To change the verification timeout, enter a new value in the **Timeout (sec)** field. The default value is 3 seconds. The range is 1 to 30 seconds. Tap the up or down arrows next to the list to change the value by 1 second in either direction.
- To change the trigger key for the application, choose **Screen Button**, **Enter**, **Scan** (default), **VolumeDown**, or **VolumeUp** from the **Trigger Key** list.

  If you choose **ScreenButton**,  appears in all screens where you can scan an address. Tap the button to turn on the imager.
- To change databases, tap a name in the **Database** list:

  The default is **verdex.db.usa.Demo** for the United States database.

Tap **verdex.db.uk.Demo** for the United Kingdom database. Use the DemoAddresses_GBR.pdf sample address file if you choose this database.

Tap **verdex.db.deu.Demo** for the Germany database. Use the DemoAddresses_DEU.pdf sample address file if you choose this database.

- To automatically print a PDF417 address label to the default Bluetooth printer when the address is verified, check the **Automatically Print Label** check box. For more information, see the next section, **"Printing Address Labels."**
- To change the Profile used for processing images, tap the **Profiles** tab and choose **Defaults** or **Parcels**.

  The Defaults profile is optimized for most address formats. The Parcels profile is optimized for formats that may include separating lines, such as formats used by commercial overnight delivery services.

3 To save your settings, tap **Save**.

  Or, to discard your changes, tap **Cancel**.

Printing Address Labels

Once you have verified an address, you can print a PDF417 address label on a PB22, PB32, or PB50 printer.

To print an address label

1 Make sure the printer is turned on.

2 In the VERDEX Results screen, tap **Print**. The label prints.

  If you have not yet chosen a printer, you are prompted to set up a connection. For help, see the next section, **"Connecting to the Mobile Printer" on page 19.**

Note: If you configured the Mobile Component to send the label to the printer when the address is verified, the printer prints the label immediately. For help, see the previous section, **"Configuring Settings."**

Connecting to the Mobile Printer

The Mobile Component can print address labels on an Intermec mobile printer running Fingerprint. Supported printers include the PB22, PB32, and PB50.

If you chose the Automatically Print Label option in the Settings, a PDF417 address label is automatically printed when the address is verified.

To print address labels, you need to connect to the mobile printer as follows:

- Scan the Bluetooth address bar code on the printer.
- Or, choose the printer from a list of discovered Bluetooth devices.

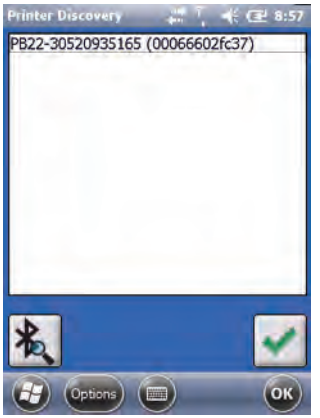
Connecting by Scanning a Bar Code

1. Tap **Menu** > **Select Printer**. The Select Printer screen appears.
2. Scan the Bluetooth address bar code on the printer. The printer device name and Bluetooth address appear in the **Current Bluetooth Printer** field.

Tap **Test** to print a printer configuration summary test label.

Connecting by Bluetooth Discovery

1. Make sure the printer is discoverable and turned on.
2. Tap  . When the printer is discovered, the Printer Discovery screen appears with the printer Bluetooth device name and address in the list.



3. Tap the printer name in the list and then tap  . When the printer is paired, a confirming message appears.
4. Tap **ok** to clear the message. The Select Printer screen appears with the selected printer name and address in the **Current Bluetooth Printer** field.
5. Tap **OK**. The Scan Address or VERDEX Results screen appears.

About Label Templates

When you install the VERDEX Mobile Component, label template files are placed in the \Program Files\Intermec\VERDEX Demo directory on the mobile computer.

The application uses these templates to print correctly formatted PDF417 address labels on the Bluetooth printer. Each template is associated with one of the demo databases and with one of the supported Intermec mobile printers, and is written in Intermec Fingerprint printer language. Templates must be located in the same directory as the Mobile Component application.

You can create your own label templates to use with the Mobile Component. Templates should be named as follows:

`[PrinterModel]_[DatabaseName].prn`

where:

PrinterModel is the Intermec printer model you are using.

DatabaseName is the name of the database to be associated with the template.

All tags in the template file use the format `VERDEX[fieldname]`, where *fieldname* corresponds to a field name in the associated database.

Note: In these sample templates, all indented code lines continue from the previous line without a manual line break.

Sample Template for PB22

This template is named `PB22_verdex.db.USA.demo.prn`.

```
CLIP OFF
CLIP BARCODE OFF
XORMODE OFF
AN 7
NASC -2
MAG 1,1:DIR 2
PP 325,690:FT "Swiss 721 BT",9,0,100
NI:PT "VERDEX[HouseNumber] VERDEX[PreDirectional]
   VERDEX[Street] VERDEX[Suffix]"
PP 295,690:FT "Swiss 721BT",9,0,100
NI:PT "VERDEX[City] VERDEX[State] VERDEX[Zip5]"
PP 325,200
BARSET "PDF417",1,1,2,4,0,1,2,0,1,0
BF OFF
PB "VERDEX[HouseNumber] VERDEX[PreDirectional]
   VERDEX[Street] VERDEX[Suffix]"+CHR$(10)+CHR$(13)+
   "VERDEX[City] VERDEX[State] VERDEX[Zip5]"
PF
```

Sample Template for PB32

```
CLIP OFF
CLIP BARCODE OFF
XORMODE OFF
AN 7
NASC -2
MAG 1,1:DIR 1
PP 10,130:FT "Swiss 721 BT",8,0,100
NI:PT "VERDEX[HouseNumber] VERDEX[PreDirectional]
   VERDEX[Street] VERDEX[Suffix]"
PP 10,99:FT "Swiss 721BT",8,0,100
NI:PT "VERDEX[City] VERDEX[State] VERDEX[Zip5]"
PP 360,135
REM BARSET "PDF417",1,1,2,5,0,1,2,0,1,0
BARSET "PDF417",1,1,2,5,0,1,2,0,1,0
BF OFFPB "VERDEX[HouseNumber] VERDEX[PreDirectional]
   VERDEX[Street] VERDEX[Suffix]"+CHR$(10)+CHR$(13)+
   "VERDEX[City] VERDEX[State] VERDEX[Zip5]"
PF
```

Sample Template for PB50

```
CLIP OFF
CLIP BARCODE OFF
XORMODE OFF
AN 7
NASC -2
MAG 1,1:DIR 3
PP 775,75:FT "Swiss 721 BT",10,0,100
NI:PT "VERDEX[HouseNumber] VERDEX[PreDirectional]
   VERDEX[Street] VERDEX[Suffix]"
PP 775,105:FT "Swiss 721BT",10,0,100
NI:PT "VERDEX[City] VERDEX[State] VERDEX[Zip5]"
PP 255,75
BARSET "PDF417",1,1,2,4,0,1,2,0,1,0
BF OFF
PB "VERDEX[HouseNumber] VERDEX[PreDirectional]
   VERDEX[Street]VERDEX[Suffix]"+CHR$(10)+CHR$(13)+
   "VERDEX[City] VERDEX[State] VERDEX[Zip5]"
PF
```

For more information on the tags used in each template, see the *Fingerprint Command Reference Manual*.

Choosing Databases

The VERDEX Mobile Component ships with demonstration databases you can use for testing and development without a license. When you purchase a license, you can add your own databases and configure the application to use them. To add databases, you need to add them programmatically when you call the Mobile Component. For information, see the next section, **"Calling VERDEX from Another Application."**

Calling VERDEX from Another Application

To call VERDEX from within another application using an indirect reference, you need to include Intermec.Components.dll.

To call VERDEX from within another application using a direct reference, you need to include VerdexComponent.exe.

The Start method for the VERDEX component takes an XML string, or a string representing the path and filename of an XML file containing property information.

Application developers can also choose to set properties on the component before calling the Start method.

About the Features Property

Features determine the kind of OCR used by VERDEX in the application. Currently, the only supported Feature is OCR of machine printed postal addresses, specified by "verdex.sw.machineprint".

When using these as parameters, list them as a "Features" list of "Feature" strings—one string representing the feature name and a second string representing the feature version.

Features can be listed in any order, but within the parameter list for each feature, the feature name must be first and the version second. Feature parameters are described in the next table.

Feature XML Parameter Descriptions

Name	Valid Values	Description
Name	String representing the name of the feature to be supported. For this release, set to "verdex.sw.machineprint".	Required. Calling application needs to specify.
Version	String representing the feature version in the format X.XX.	Required. Calling application needs to specify.

About the Databases Property

Databases contain the postal address information VERDEX uses to confirm the address. When using these as parameters, list them as a "Databases" list of "Database" strings—one string representing the database name, a second string representing the database version, and a third string representing the path to and name of the database configuration file. Multiple databases are allowed.

Databases can be listed in any order, but within the parameter list for each database, the database name must be first, the version second and the configuration file path and name must be third. Database parameters are described in the next table.

Database XML Parameter Descriptions

Name	Valid Values	Description
Name	String representing the database name.	Required.
Version	String representing the database version in the format X.XX.	Required.
Path	String representing the file name and path to the database configuration file.	Required. This .ini file includes database information.

Note: If no database is specified, the VERDEX Component runs in demo mode and uses the demo databases.

About the ConfigFileName Property

ConfigFileName is a string representing the path to and name of the configuration file. The default path is to the application directory. The path to the configuration file must be passed in to the component.

ConfigFile XML Parameter Description

Name	Valid Values	Description
Path	String representing the path and filename of the VERDEX configuration file.	If no file is specified, the default is used. If you change this in the application settings screen, this XML is updated to match.

VERDEX Property XML Example

This example shows how to list the Database, Feature, and ConfigFile properties:

```
<?xml version="1.0" encoding="utf-8" ?>
<VERDEXParameters>
   <Databases>
      <Database>
         <Name>verdex.db.deu.demo</Name>
         <Path>\Program Files\Intermec\VERDEX\Databases
          \Germany.ini</Path>
         <Version>1.0</Version>
      </Database>
   </Databases>
   <Features>
      <Feature>
         <Name>verdex.sw.machineprint</Name>
         <Version>1.0</Version>
      </Feature>
   </Features>
   <ConfigFile>Path="\Program Files\Intermec\VERDEX Demo
    \Profiles\Defaults.ini"</ConfigFile>
</VERDEXParameters>
```

C# Sample Code

These samples illustrate how to assign properties and call the VERDEX component.

Note: In these code samples, all indented code lines continue from the previous line without a manual line break.

Sample 1: With Known Component Name and Indirect Reference

```
Using Intermec.Components;
public static void Main()
{
  IComponent verdexComponent =
    Intermec.Components.ReflectionHelper.CreateComponent("VERDEX");

  if (verdexComponent != null)
  {
    if (verdexComponent.Supported)
    {

    // see xml file example for what this should look like
      string verdexInput  = "<?xml version="1.0" encoding="utf-8"?>";
      verdexInput += "<VERDEXParameters>";
      verdexInput += "<Databases>";
      verdexInput += "<Database>";
      verdexInput += "<Name>verdex.db.deu.demo</Name>";
      verdexInput += "<Path>\\Program
        Files\\Intermec\\VERDEX\\Databases\\Germany.ini</Path>";
      verdexInput += "<Version>1.0</Version>";
      verdexInput += "</Database>";
      verdexInput += "</Databases>";
      verdexInput += "<Features>";
      verdexInput += "<Feature>";
      verdexInput += "<Name>verdex.sw.machineprint</Name>";
      verdexInput += "<Version>1.0</Version>";
      verdexInput += "</Feature>";
      verdexInput += "</Features>";
      verdexInput += "<ConfigFile Path=\"\\Program Files\\Intermec\\VERDEX
        Demo\\VerdexOcr.ini\"/>”;
      verdexInput += "</VERDEXParameters>";

      verdexComponent.Start(new object[] {verdexInput});
    }
  }
}
```

Sample 2: Using a Direct Reference

```
public static void Main()
{
  VerdexComponent.VerdexController verdexComponent = new
    VerdexComponent.VerdexController();

  // see xml file example for what this should look like
    string verdexInput  = "<?xml version="1.0" encoding="utf-8"?>";
    verdexInput += "<VERDEXParameters>";
    verdexInput += "<Databases>";
    verdexInput += "<Database>";
    verdexInput += "<Name>verdex.db.deu.demo</Name>";
    verdexInput += "<Path>\\Program
      Files\\Intermec\\VERDEX\\Databases\\Germany.ini</Path>";
    verdexInput += "<Version>1.0</Version>";
    verdexInput += "</Database>";
    verdexInput += "</Databases>";
    verdexInput += "<Features>";
    verdexInput += "<Feature>";
    verdexInput += "<Name>verdex.sw.machineprint</Name>";
    verdexInput += "<Version>1.0</Version>";
    verdexInput += "</Feature>";
    verdexInput += "</Features>";
    verdexInput += "<ConfigFile Path=\"\\Program Files\\Intermec\\VERDEX
      Demo\\VerdexOcr.ini\"/>";
    verdexInput += "</VERDEXParameters>";

    verdexComponent.Start(new object[] {verdexInput});

}
```

Sample 3: Using Properties With a Direct Reference

```
public static void Main()
{
  VerdexComponent.VerdexController verdexComponent = new
    VerdexComponent.VerdexController();

  List<List<string>> databases = new List<List<string>>;
  List<List<string>> features = new List<List<string>>;

  //Add the first database to the list of databases
  List<string> database1 = new List<string>;
  // add the database name
  database1.Add("verdex.db.deu.demo");
  //add the database version
  database1.Add("1.05");
  //add the database configuration file
  database1.Add("\\Program Files\\VERDEX\\verdex.db.deu.demo.ini");
  //add the database to the databases list
  databases.Add(database1);

  //Add the second database to the list
  List<string> database2 = new List<string>;
  // add the database name
  database2.Add("verdex.db.usa.demo");
  //add the database version
  database2.Add("1.02");
  //add the database configuration file
  database2.Add("\\Program Files\\VERDEX\\verdex.db.usa.demo.ini");
  //add the database to the databases list
  databases.Add(database2);

  //add a feature to the features list
  List<string> feature1 = new List<string>;
  //add the feature name
  feature1.Add("verdex.sw.machineprint");
  //add the feature version
  feature1.Add("1.0");
  features.Add(feature1);

  verdexComponent.Features = features;
  verdexComponent.Databases = databases;
  verdexComponent.ConfigFileName = "\\Program
    Files\\VERDEX\\VERDEXOCR.ini";
  verdexComponent.Start(null);

}
```

Sample 4: With Executable Name and Path

```
public static void Main()
{
  System.Diagnostics.Process verdexComponent = new
    System.Diagnostics.Process();
  verdexComponent.StartInfo.FileName = @"\Program
    Files\Intermec\VERDEX Demo\VerdexComponent.exe";

  string path = @"\Program Files\Intermec\VERDEX Demo\MySettings.xml";
  System.IO.StreamWriter writer = new System.IO.StreamWriter(path);

  writer.WriteLine("<?xml version=\"1.0\" encoding=\"utf-8\"?>");
  writer.WriteLine("<VERDEXParameters>");
  writer.WriteLine("<Databases>");
  writer.WriteLine("<Database>");
  writer.WriteLine("<Name>verdex.db.deu.demo</Name>");
  writer.WriteLine("<Path>HYPERLINK "file:///\\Program" \\Program
    Files\\Intermec\\VERDEX Demo\\Databases\\Germany.ini</Path>");
  writer.WriteLine("<Version>1.0</Version>");
  writer.WriteLine("</Database>");
  writer.WriteLine("</Databases>");
  writer.WriteLine("<Features>");
  writer.WriteLine("<Feature>");
  writer.WriteLine("<Name>verdex.sw.machineprint</Name>");
  writer.WriteLine("<Version>1.0</Version>");
  writer.WriteLine("</Feature>");
  writer.WriteLine("</Features>");
  writer.WriteLine("<ConfigFile Path=\"<\\Program Files\\Intermec\\VERDEX
    Demo\\VerdexOcr.ini\>"/>");
  writer.WriteLine("</VERDEXParameters>");

  writer.Close();
  writer = null;

  verdexComponent.StartInfo.Arguments = path;
  verdexComponent.Start();
}
```

Return Values

The VERDEX application component returns an XML string to the calling application with information from all addresses that were scanned during the VERDEX session.

- If an address was matched exactly, VERDEX returns that address. If a partial match resulted in multiple addresses, VERDEX returns the address that was selected by the user as seen in this example:

```
<outputs>
   <output>
      <records>
         <record>
            <field name="ThoroughfareName">Ludwigstr
             </field>
            <field name="ThoroughfareNumber">2</field>
            <field name="Town">Freiburg</field>
            <field name="Postcode">79104</field>
            <field name="Country">Germany</field>
         </record>
      </records>
   </output>
</outputs>
```

- If the address returned from the database was incorrect and the user edited the value, then the edited version of the address is also returned:

```
<outputs>
   <output>
      <records>
         <record>
            <field name="Address1">446 Oak Street NE
             </field>
            <field name="Address2">Apt 1</field>
            <field name="City">Everett</field>
            <field name="State">WA</field>
            <field name="Postcode">79104</field>
            <field name="Country">USA</field>
         </record>
      </records>
   </output>
</outputs>
```

- If the address did not result in any matches, the raw OCR results for that address are returned:

```
<output>
   <records>
      <record>
         <ocr xmlns="http://www.raf.com/verdex">
            <line>
               <word>
                  <char>
                     <possibility char="X" conf="239" />
                     <possibility char="x" conf="238" />
                  </char>
                  <char>
                     <possibility char="M" conf="255" />
                     <possibility char="N" conf="82" />
                     <possibility char="H" conf="60" />
                  </char>
                  <char>
                     <possibility char="L" conf="225" />
                  </char>
               </word>
               <word>
                  <char>
                     <possibility char="t" conf="247" />
                  </char>
                  <char>
                     <possibility char="a" conf="235" />
                     <possibility char="8" conf="82" />
                  </char>
                  <char>
                     <possibility char="g" conf="239" />
                  </char>
                  <char>
                     <possibility char="5" conf="226" />
                  </char>
               </word>
            </line>
         </ocr>
      </record>
   </records>
</output>
```

- If an error occurs and an address is not verified, the returned XML may be as follows:

```
<output>
   <error>ERROR TEXT HERE<error>
</output>
```

VERDEX Property Descriptions

If you choose not to use properties, the VERDEX component takes a string of XML. These settings can be configured using the SmartSystems API (SSAPI) or by using Intermec Settings on the computer.

- **Allow Print** is a boolean value specifying whether the user should be able to print labels.
- **Automatic Print** is a boolean value specifying whether to print an address label automatically after verification.
- **Manual Entry** is a boolean value indicating that the calling application wants to go directly to the Manual Entry view in the component.
- **Results FileName** is the path to and name of the file in which the component writes the results. If this is blank, the results are returned in a string of XML.
- **Timeout** is a numeric value representing the maximum amount of time to verify an address before returning a timeout error. Range is .5 to 30 seconds.
- **Trigger Key** is a string representing the hardware key used to trigger the VERDEX process. Valid values include: None, Scan, Enter, VolumeUp, and VolumeDown.
- **Return All Results** is a boolean value specifying whether results from all scans are returned. If set to false, only results from the last scan are returned to the calling application or written to the results file.
- **Return Without Displaying** is a boolean specifying whether to return to the calling application without displaying the address within the component.

For examples, see the sample code provided with the VERDEX Solution Component.

Intermec

Worldwide Headquarters
6001 36th Avenue West
Everett, Washington 98203
U.S.A.

tel 425.348.2600

fax 425.355.9551

www.intermec.com

VERDEX Mobile Component User's Guide

P/N 934-059-002